Meta Architecture for Point Cloud Analysis

Haojia Lin[1], Xiawu Zheng[2], Lijiang Li[1], Fei Chao[1], Shanshan Wang[2], Yan Wang[3], Yonghong Tian[2], Rongrong Ji*[1,2]

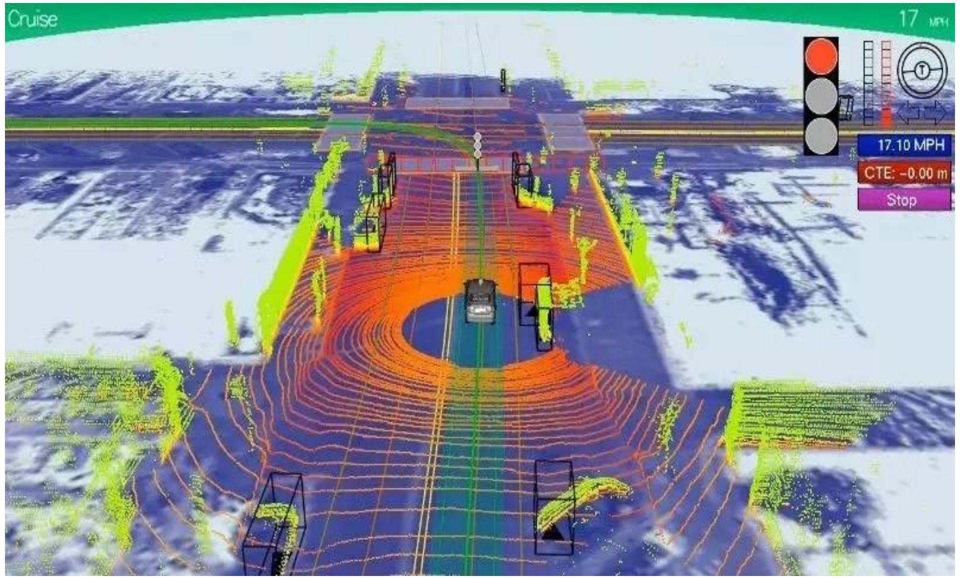
[1]Media Analytics and Computing Lab, Department of Artificial Intelligence, School of Informatics, Xiamen University, China. [2]Peng Cheng Laboratory. [3]Samsara Inc.

Contact: rrji@xmu.edu.cn

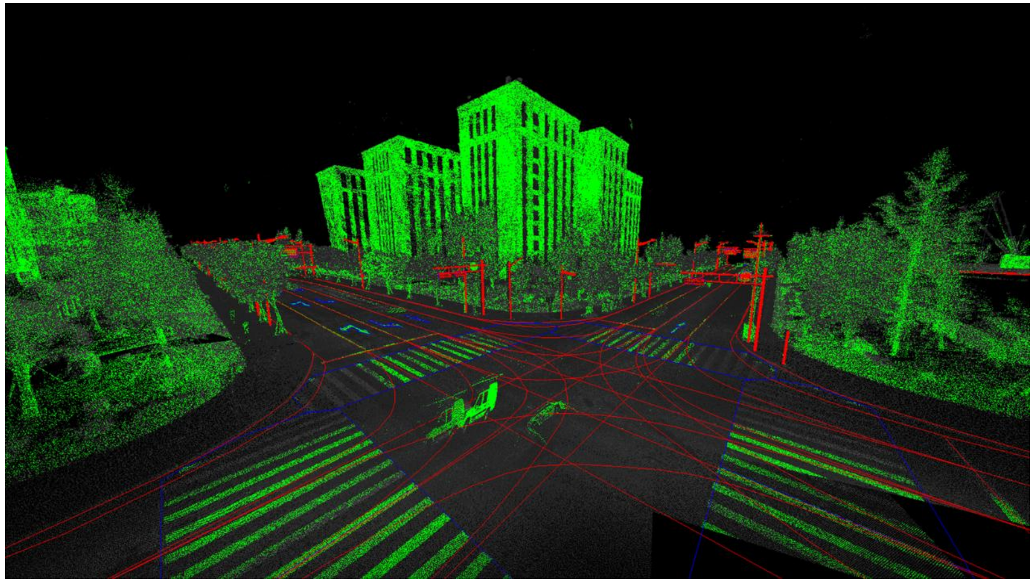
THU-AM-115

Point Cloud

- Sparse
- Unordered
- Irregular



感知



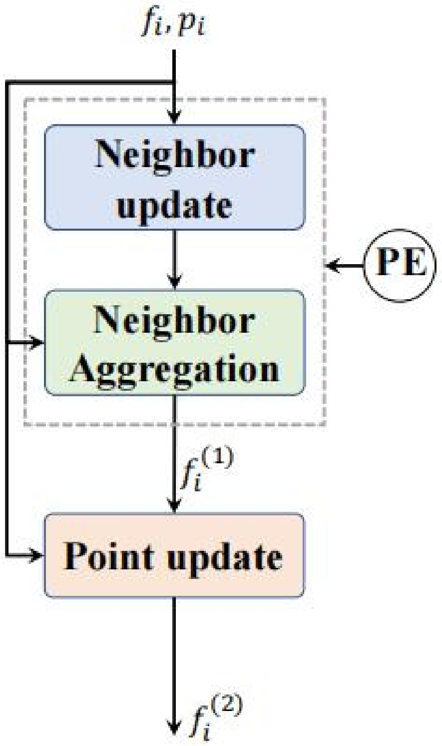
重建

Existing Methods

- PointNet (2017)
- PointNet++ (2017)
- DGCNN (2018)
- PointCNN (2018)
- Point Transformer (2021)
- ASSANet (2021)
- PointNeXt (2022)

PointMeta: Meta Architecture for Point Cloud Analysis

- 4 Meta Functions
  - Neighbor Aggregation
  - Neighbor Aggregation
  - Point Update
  - Position Embedding



(a) PointMeta

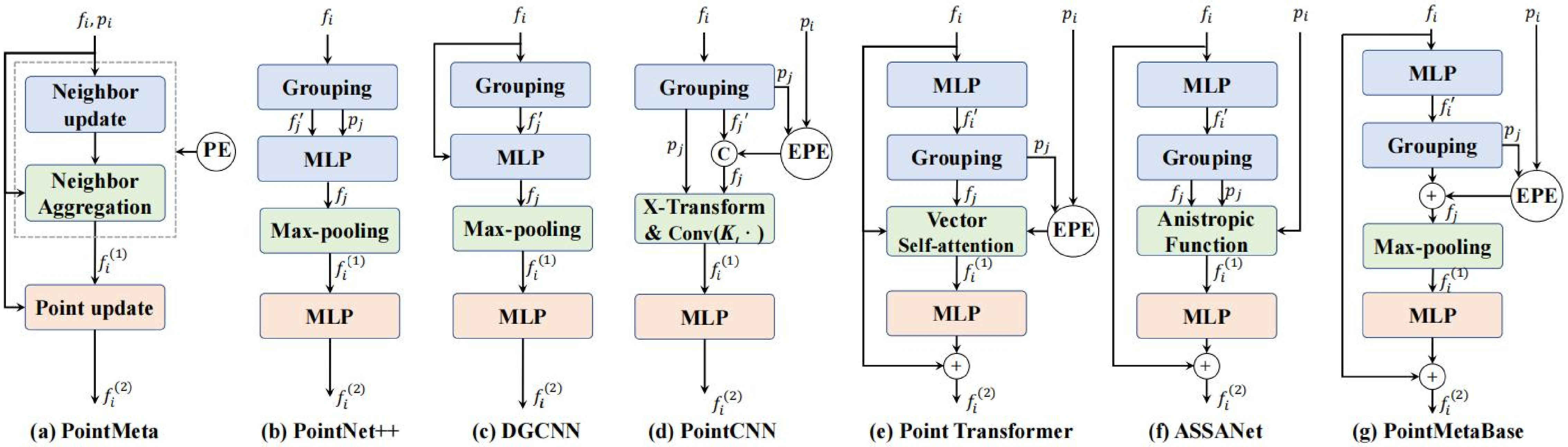
$$\boldsymbol{f}_{\mathcal{N}(i)} = \phi^n \circ \phi^e(f_i, p_i), \tag{1}$$

$$f_i^{(1)} = \phi^a \circ \phi^e(f_i, p_i, \boldsymbol{f}_{\mathcal{N}(i)}, \boldsymbol{p}_{\mathcal{N}(i)}) \tag{2}$$

$$f_i^{(2)} = \phi^p(f_i^{(1)}, p_i), \tag{3}$$

PointMeta

- PointMeta and its instantiation examples



Empirical Study

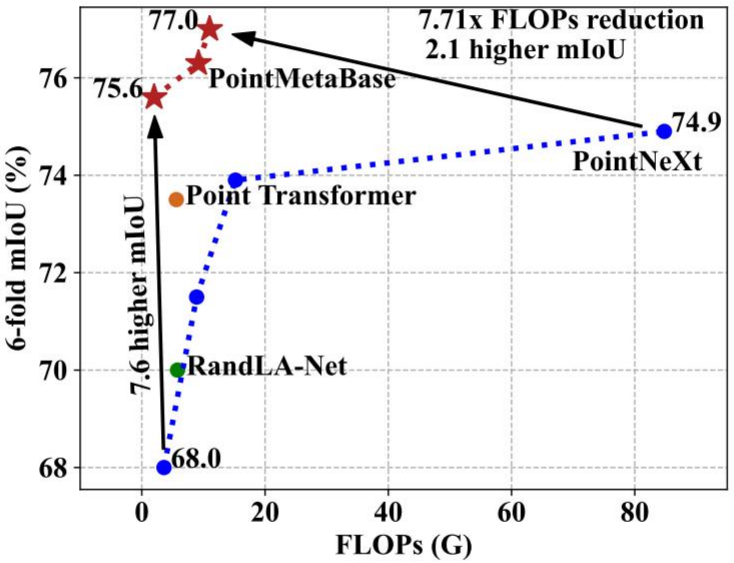
- Best Practice
  - Position Embedding
  - Neighbor Aggregation
  - Complexity Allocation

Variant	mIoU (%)	Params (M)	FLOPs (G)
Plain-Max	47.3±0.7	2.0	1.4
Plain-PP	65.1±0.2	2.0	1.5
Plain-PP-Max	58.4±0.6	2.0	1.5
Plain-IPE-Max	68.0±0.3	2.0	13.8
Plain-EPE-Max	**69.0±0.3**	2.0	1.8
Plain-EPE-PP	65.4±0.1	2.0	1.8

Variants	mIoU (%)	Params (M)	FLOPs (G)
N1P1	69.0±0.3	2.0	1.7
N2P0	68.7±0.3	2.0	1.7
N0P2	67.8±0.4	2.0	1.7
N1P2	69.5±0.3	2.7	2.0
N1P2-Inv	**69.7±0.3**	7.1	3.8
N2P1	69.4±0.3	2.7	2.0
N2P1-Inv	69.6±0.4	7.1	3.8
N1P3	69.5±0.3	3.4	2.3
N3P1	69.6±0.5	3.4	2.3

Variants	mIoU (%)	Params (M)	FLOPs (G)
Point Trans [36]	70.5±0.3	7.8	5.6
Point Trans (-VSA+Max)	70.3±0.2	5.1	3.3

Evaluation



Method	S3DIS 6-Fold mIoU (%)	S3DIS 6-Fold OA (%)	S3DIS Area-5 mIoU (%)	S3DIS Area-5 OA (%)	ScanNet mIoU Val (%)	ScanNet mIoU Test (%)	Params. M	FLOPs G	Throughput (ins./sec.)
PointNet++ [14]	54.5	81.0	53.5	83.0	53.5	55.7	1.0	7.2	237
PointCNN [8]	65.4	88.1	57.3	85.9	-	45.8	0.6	-	-
DeepGCN [7]	60.0	85.9	52.5	-	-	-	3.6	-	-
KPConv [22]	70.6	-	67.1	-	69.2	68.6	15.0	-	-
RandLA-Net [5]	70.0	88.0	-	-	-	64.5	1.3	5.8	-
BAAF-Net [17]	72.2	88.9	65.4	88.9	-	-	5.0	-	-
Point Transformer [32]	73.5	90.2	70.4	90.8	70.6	-	7.8	5.6	-
CBL [20]	73.1	89.6	69.4	90.6	-	70.5	18.6	-	-
ASSANet [15]	-	-	65.8	88.9	-	-	2.4	2.5	300
ASSANet-L [15]	-	-	68.0	89.7	-	-	115.6	36.2	136
PointNeXt-S [16]	68.0	87.4	63.4±0.8	87.9±0.3	64.5	-	0.8	3.6	276
PointNeXt-B [16]	71.5	88.8	67.3±0.2	89.4±0.1	68.4	-	3.8	8.9	161
PointNeXt-L [16]	73.9	89.8	69.0±0.5	90.0±0.1	69.4	-	7.1	15.2	109
PointNeXt-XL [16]	74.9	90.3	70.5±0.3	90.6±0.1	71.5	71.2	41.6	84.8	43
PointMetaBase-L	**75.6**	**90.6**	69.5±0.3	90.5±0.1	71.0	-	2.7	2.0	187
PointMetaBase-XL	**76.3**	**91.0**	**71.1±0.4**	**90.9±0.1**	71.8	-	15.3	9.2	104
PointMetaBase-XXL	**77.0**	**91.3**	**71.3±0.7**	**90.8±0.6**	72.8	71.4	19.7	11.0	90

Conclusion

- In the dimension of models, it allows us to compare and contrast different models in a fair manner.
- In the dimension of components, it allows us to have a higher level view across components.
- Based on the learnings from the previous two dimensions, we are then able to do simple tweaks on the building blocks to apply the best practices.

Thanks!